(Applied) Cryptography

Tutorial #6

Manuel Barbosa (mbb@fc.up.pt) Rogério Reis (rogerio.reis@fc.up.pt)

November 18, 2022

1. In a public-key system using RSA, you intercept the ciphertext $C = 20$ sent to a user whose public key is $e = 13$, $n = 77$. What is the plaintext M?

2. In a RSA system, the public key of a given user is $e = 65$, $n = 2881$. What is the private key of this user?

3. In the RSA public-key encryption scheme, each user has a public key e and a private key d. Suppose Bob leaks his private key. Rather than generating a new modus, he decides to generate a new public key e and a new private key d. Is this safe?

4. Suppose Bob uses the RSA cryptosystem with a very large modulus n for which the factorisation cannot be found in a reasonable amount of time. Suppose Alice sends an enciphered message to Bob containing only her phone number: $\mathsf{number}^e \pmod n$. Is this safe?

5. Although, since 2002, there is a published algorithm with polynomial complexity to test primality of an integer, its performance for small sizes is too slow to be considered as usable. What is normally used is a probabilistic test, that can be iterated the necessary number of times so that the probability of a false positive may be made negligible. The Miller-Rabin is a primality test of this kind.

   Theorem 1. *If p is an odd prime, then the equation*

   $$x^2 \equiv 1 \pmod p$$

   has only two solutions: $x \equiv 1$ and $x \equiv -1$.

   Proof. If x is solution of the equation, then

   $$\begin{aligned} x^2 - 1 &\equiv 0 \pmod p \\ (x+1)(x-1) &\equiv 0 \pmod p \end{aligned}$$

   thus

   $$p \mid (x+1) \vee p \mid (x-1).$$

   Suppose that $p \mid (x+1) \wedge p \mid (x-1)$. Then we can write $(x+1) = kp$ and $(x-1) = jp$ for some integers k and j. Subtracting both equations we get $2 = (k-j)p$ that is only satisfied with $p = 2$, but the initial assumption states that p is an odd prime. Thus $p \mid (x+1) \dot{\vee} p \mid (x-1)$. Suppose that $p \mid (x-1)$. Then

   $$(\exists k)(x - 1 = kp)$$

   and hence $x \equiv 1 \pmod p$.

   In an entirely analogous manner we proceed if $x \equiv -1 \pmod p$. □

   We can look at this theorem in a different perspective: if we can find a solution for $x^2 \equiv 1 \pmod n$ that is different from $x = \pm 1$, then we can conclude that n is not prime.

Theorem 2. *Let p be an odd prime and a such that $p \nmid a$. We can always express $p-1$ as*

$$p - 1 = 2^k d$$

with d odd. Thus, one of the two following is true:

(a) $a^d \equiv 1 \pmod{p}$,

(b) $\exists i \in \{0, \ldots, k-1\}\ a^{2^i d} \equiv -1 \pmod{p}$.

Proof. By Fermat's theorem, $a^{2^k d} \equiv 1 \pmod{p}$. Thus, in the following sequence

$$a^d, a^{2d}, a^{2^2 d}, \ldots, a^{2^k d}$$

at least the last is congruent with 1. But each of the powers of a is the square of the previous. Thus, one of the following is true

(a) $a^d \equiv 1 \pmod{p}$;

(b) $\exists i \in \{1, \ldots, k\}$,
$$a^{2^i d} \equiv 1 \pmod{p} \ \wedge\ a^{2^{i-1} d} \not\equiv 1 \pmod{p}.$$

As we are in the conditions of the previous theorem, we conclude that

$$a^{2^{i-1} d} \equiv -1 \pmod{p}.$$

□

We can, then, write a programming function, WITNESS, that takes a number n and a "witness" a, with $(a, n) = 1$, and tests if $a^d \not\equiv 1 \pmod{n}$ and $a^{2^i d} \not\equiv -1 \pmod{n}$, for all $0 \le i \le k$. If the test succeeds we know for sure that the number is not a prime. If it fails we cannot conclude, but we have a probability of $\frac{1}{2}$ of n being a prime. We can repeat the test (with a different values for a). If we try m times and all the tests are negative we can ensure that the number n is a prime with a probability $1 - 2^{-m}$.

Programming assignment: Write a python program that implements this strategy and test it for large primes.